(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 December 2000 (28.12.2000)

PCT

(10) International Publication Number
WO 00/79484 A1

(51) International Patent Classification⁷: G06T 13/00

(21) International Application Number: PCT/AU00/00694

(22) International Filing Date: 21 June 2000 (21.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PQ 1087 21 June 1999 (21.06.1999) AU

(71) Applicant *(for all designated States except US)*: **UNISEARCH LIMITED** [AU/AU]; Rupert Myers Building, Level 2, Gate 14, Barker Street, UNSW, Sydney, NSW 2052 (AU).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **JIN, Jesse, Sheng** [AU/AU]; Rupert Myers Building, Level 2, Gate 14, Barker Street, UNSW, Sydney, NSW 2052 (AU). **XIA, Jiali** [CN/AU]; Rupert Myers Building, Level 2, Gate 14, Barker Street, UNSW, Sydney, NSW 2052 (AU).

(74) Agent: F B RICE & CO; 605 Darling Street, Balmain, NSW 2041 (AU).

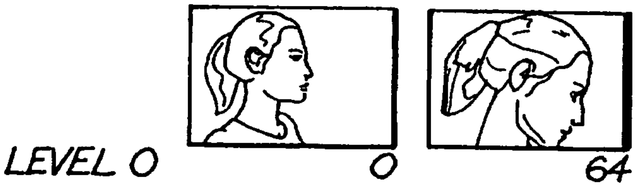
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: PROGRESSIVE MULTI-FRAME ANIMATION SOFTWARE FOR WEB OBJECTS



(57) Abstract: An animation tool suitable for use with web objects. In particular a method and system for the progressive animation of web objects. A transmitter and method of transmission, a receiver and method or reception, and a signal for transmission and reception. Strings of frames are divided from an animation sequence comprising an ordered series of frames. The first string comprising one frame of the sequence, or more than one frame widely spaced from each other in the sequence. The second string contains the same number or more frames than the first, and the frames of the second string are widely spaced from each other and from the frame, or between the frames, of the first string. Each successive string then contains the same number or more frames than the preceding string, and the frames of each successive string are widely spaced from each other and between the frames of the immediately preceding string. The strings are transmitted and on receipt, are displayed. First a first partial animation sequence comprising the frame, or frames, of the first string in a first thread are displayed repeatedly until it reaches its last frame and is killed. Following reception of the second string, a second partial animation sequence comprising the frames of the first and second string, in sequential order, are displayed in a second thread which will repeat until it reaches its last frame and is killed following reception of the next string. Then, once each next string is received and the last thread reaches its last frame, the last thread is killed and a next animation sequence comprising all the frames of the received strings, in sequential order, is displayed in a next thread and so on until all the strings are received and the entire animation sequence is being displayed.

WO 00/79484 A1

Title

Progressive multi-frame animation software for web objects

Technical Field

This invention concerns an animation tool suitable for use with web objects. In particular the invention is a method and system for the progressive animation of web objects. In other aspects the invention also concerns a transmitter and method of transmission, a receiver and method of reception, and a signal for transmission and reception.

A further invention involves non-linear replay of animation at the client side. This may be used where the client side parts of the primary invention are present, regardless of whether the animated sequence is being transmitted or recovered from memory.

Background Art

In spite of rapid development in multimedia communication technology, the ability to efficiently deliver multimedia objects over the network is still a problem. A widely used technique on the Internet is the multi-frame structure and linear replay mechanism of the GIF file. In this technique an entire file is loaded and then animation is started. The animation then has a linear replay mode. The animation is not available for other file formats, the GIF format is not compressed and individual frames cannot be indexed and referenced.

Many researches have been done on delivering video over the Internet. The techniques range from video server design, delivery agent, adaptive coding of video and low-level network protocols. Proactive buffer management for delivery of video stream using *a priori* information stored in the video stream has also been proposed. This scheme also needs to monitor the available bandwidth. These techniques are all either too expensive or not suitable for animation.

Summary of the Invention

Progressive Animation for Web Objects

A first aspect of this invention is a method for animating web objects, including the following steps:

(a) Dividing an animation sequence comprising an ordered series of frames, into a plurality of strings of different frames. The first string comprises one frame of the sequence, or more frames widely spaced from each other in the sequence. The second string contains one or more frames, and the frames of the second string are widely spaced from each other and from the frame, or between the frames, of the first string. Each successive string then contains the same number or more frames than the preceding strings, and the frames of each successive string are widely spaced from each other and between the frames of the immediately preceding string.

(b) Transmitting the first string of frames.

(c) Receiving the first string of frames and then, using multi-thread controls, displaying a first partial animation sequence comprising the frame, or frames, of the first string in a first thread which will repeat until it is killed.

(d) Transmitting and receiving the second string of frames while the first partial animation sequence is being displayed.

(e) Then, once the second string is received and the first thread reaches its last frame, killing the first thread and displaying a second partial animation sequence comprising the frames of the first and second string, in sequential order, in a second thread which will repeat until it is killed.

(f) transmitting and receiving the next string of frames while the last partial animation sequence is being displayed.

(g) Then, once the next string is received and the last thread reaches its last frame, killing the last thread and displaying a next animation sequence comprising all the frames of the received strings, in sequential order, in a next thread which will repeat until it reaches its last frame and is killed.

(h) Repeating steps (f) and (g) until all the strings have been received and the entire animation sequence is being displayed.

A second aspect of this invention, as currently envisaged, is a system for animating web objects, including:

(a) Dividing means to divide an animation sequence comprising an ordered series of frames, into a plurality of strings of different frames. Where, the first string comprises one frame of the sequence, or more than one frame widely spaced from each other in the sequence. The second string contains the same number or more frames than the first, and the frames of the second string are widely spaced from each other and from the frame, or between the frames, of the first string. Each successive string then contains the same number or more frames than the preceding string, and the frames of each successive string are widely spaced from each other and between the frames of the immediately preceding string.

(b) A transmitter to transmit the strings of frames in order, starting with the first.

(c) A receiver to receive the strings of frames in order, starting with the first.

(d) A multi-thread controller to display a first partial animation sequence comprising the frame, or frames, of the first string in a first thread which will repeat until it reaches its last frame and the controller kills it following reception of the second string. The controller will then display a second partial animation sequence comprising the frames of the first and second string, in sequential order, in a second thread which will repeat until it reaches its last frame and is killed following reception of the next string. Then, once each next string is received and the last thread reaches its last frame, the controller kills the last thread and displays a next animation sequence comprising all the frames of the received strings, in sequential order, in a next thread and so on until all the strings are received and the entire animation sequence is being displayed.

In further aspects the invention is a transmitter and method of transmission, a receiver and method of reception, and a signal for transmission and reception, which each involve the appropriate features mentioned above. The receiver may employ client-side buffers, progressive loading frames, and multi-thread animation. The signal itself is divided into 'pyramid' structured stages. The top layer of the 'pyramid' is first transmitted with as small amount of data as possible and then the rest of the layers are transmitted progressively through each level.

The invention enjoys the advantage that some, albeit crude, animation is displayed very quickly after transmission begins since the first string has only a relatively small amount of data and is transmitted relatively quickly. The quality of the animation is refined in steps as each successive string arrives, and no jitter appears in the animation. Since each successive string will usually have more data than the last, each step in improvement usually takes longer than the last until the entire animation sequence is being displayed. This technique can give users a better view of the whole animation structure as early as possible so the user can decide to terminate the transmission at certain points. It also suits multi-cast as animation can be performed regardless the network capacity and storage space in the client side.

Another advantage of the animator is that it does not require special coded objects. The sequence can still be in original frames, and any file format can be supported. Therefore, the single frame in the original sequence can still be referenced and used for other purposes, such as non-linear replay. The animator can be easily incorporated into Web browser or attached as a plug-in processor.

Although versatile functions are defined, the animator class is relatively compact (less than 6K). It has less overhead comparing with other methods of animation. It can be downloaded from remote site and activated locally, or pre-installed in the Netscape browser as a plug-in.

Non-Linear Control

Conventional frame-based animation uses linear replay and the speed of replay is fixed. This is not convenient and sufficient. The usual way to incorporate non-linear replay or change replay speed is to produce duplicated frames and arrange the frames in a pre-defined order. However, this will increase the size of the sequence and transmit redundant frames.

The animator described above is lightweight in the sense that it does not carry redundant frames and coding information. In this animator, the control is handled at the client end. It is easy to incorporate non-linear control into the animation. There are three ways to achieve non-linear play:

First, it is possible to relate the next frame to be displayed to the last by some formula dependant on the movement of a pointer. For instance, the display of the next frame during the nonlinear control can be expressed as: $f_n = f_c + f_{d_2 - d_1}$, where f_n denotes the next frame number, f_c is the current frame number and $(d_2 - d_1)$ can be calculated from mouse movement. Using this technique the client can control the animation speed in both reverse and forward orders by simply dragging the mouse backward or forward with different speeds. The speed of animation increases or decreases with the speed of mouse movement, while the frame number is displaying to provide a better way to view the animation.

Second, a duration tag can be attached to each frame to define the duration of each frame, say in milliseconds. There is no extra coding needed for the frames.

Third, the animation can also be controlled by a parametrical function, where a waveform, such as a sinusoid, is used to define the velocity of the animation sequence.

Brief Description of the drawings

An example of the invention will now be described with reference to the accompanying drawings, in which:

Fig. 1 is the first two frames of a sequence of seventytwo frames, that are displayed in a first partial animation.

Fig. 2 is the three frames of the sequence that are displayed in a second partial animation.

Fig. 3 is the five frames of the sequence that are displayed in a third partial animation.

Fig. 4 is the nine frames of the sequence that are displayed in a fourth partial animation.

Fig. 5 is the eighteen frames of the sequence that are displayed in a fifth partial animation.

Fig. 6 is the thirtytwo frames of the sequence that are displayed in a sixth partial animation.

Fig. 7 is the remaining thirtytwo frames of the sequence that form the seventh string of frames that complete the animation when joined to the sixth partial animation.

Best Modes of the Invention

Progressive Animation for Web Objects

The invention makes use of a client-side buffer and multi-thread task management to speed up loading and smooth animation.

For an N frame sequence, $F_0, F_1, \ldots, F_{N-1}$, the transmission starts with $n_0 = \lceil N/2^{k-1} \rceil$ frames and is progressed in a pyramid of $n_0 * 2^{k-1}$ ($k = 1, 2, \ldots, K$). During the transmission of the kth level, the frames already downloaded at levels 0 to k-1 are animated in a thread. The old thread is killed only when it is at the normal termination and a new thread has been started. Therefore, animation is refined during the loading process and no jitter appears in animation.

To organise the animation frames into the pyramid structure, let N be the total frame number of the animation sequence, then divide the N frames into K levels. The size of each level is $n_k = 2^k$ ($k = 1, 2, \ldots, K$). The combination of all levels, in order of increasing size, consists of all frames of animation. That means:

$$N = n_0 + n_1 + \ldots + n_K = \Sigma n_k$$

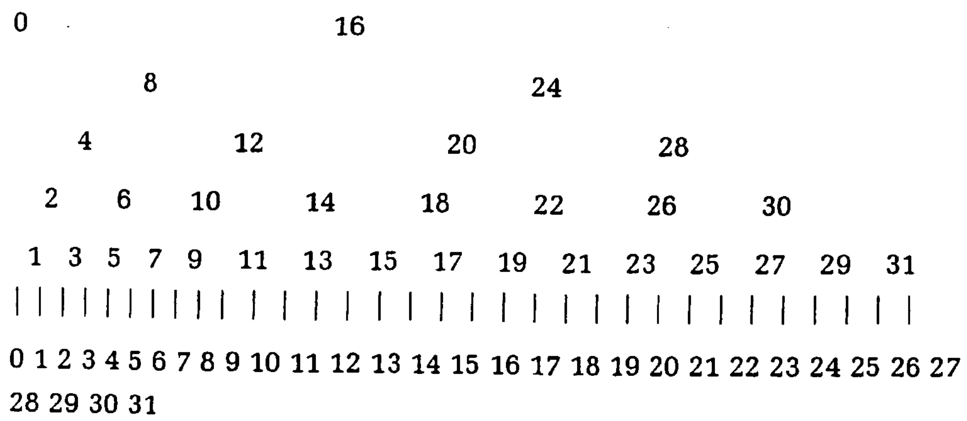
where, $n_0 = \lceil N/2^{K-1} \rceil$ defines the number of frames at the starting level.

For level i (where $i = 1, 2, \ldots, K$), the position of the first frame of that level is $f_i = 2^{K-i}$ and the steps between adjacent frames are 2^{K-i-1}. Therefore, to calculate the jth frame at the ith level, that is, f_{ij}, the following formula can be used:

$$f_{ij} = 2^{K-i} + j^* 2^{K-i-1}$$

where $0 \leq j < 2^{i+1}$ and $0 \leq f_{ij} < N$

There are two ways to define level K: $K = \lceil \log_2 N \rceil$ and $K = \lfloor \log_2 N \rfloor$. When N equals a power of 2, both methods have the same level. An example of the pyramid structure of 32 frames is illustrated below:



The structure is made available in levels of increasing sizes, hence the name *pyramid* is used.

More examples of the pyramid are listed as follows:

N=16, K=4

start : 0 8

level 1: 4 12

level 2: 2 6 10 14

level 3: 1 3 5 7 9 11 13 15

N=64, K=6

start : 0 32

level 1: 16 48

level 2: 8 24 40 56

level 3: 4 12 20 28 36 44 52 60

level 4: 2 6 10 14 18 22 26 30 34 38 42 46 50 54 58 62

level 5: 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63

Note that the number of frames do not have to be a power of 2. When it does not equal a power of 2, the two definitions of K produce different pyramids. There may also have not completely filled levels. The following two examples are from $K = \lceil \log_2 N \rceil$:

N=30, K=5

start : 0 16

level 1: 8 24

level 2: 4 12 20 28

level 3: 2 6 10 14 18 22 26

level 4: 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29

Now, corresponding to the animation illustrated in Figures 1 to 7, the following pyramid shows how a sequence of seventy two frames may be animated:

N=72, K=7

start (first string) : 0 64

level 1 (second string) : 32

level 2 (third string) : 16 48

level 3 (fourth string) : 8 24 40 56

level 4 (fifth string) : 4 12 20 28 36 44 52 60 68

level 5 (sixth string) : 2 6 10 14 18 22 26 30 34 38 42 46 50 54 58 62 66 70

level 6 (seventh string) : 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71

With $K = \lfloor \log_2 N \rfloor$, animation starts with $n_0 = \lceil N/2^{K-1} \rceil$ frames. The above two examples appear as follows. They are one level less than the previous two examples but have more starting frames. Both methods are very efficient. The former suits for applications with small network capacity but requiring fast start, and the latter suits for applications with reasonable network capacity and not critical on start speed.

N=30. K=4

start : 0 8 16 24

level 1: 4 12 20 28

level 2: 2 6 10 14 18 22 26

level 3: 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29

N=72, K=6

start : 0 32 64

level 1: 16 48

level 2: 8 24 40 56

level 3: 4 12 20 28 36 44 52 60 68

level 4: 2 6 10 14 18 22 26 30 34 38 42 46 50 54 58 62 66 70

level 5: 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71

The animation and downloading is concurrent in such a way that while downloading the frames of level *i* (*i* = 1, 2, ..., K), frames already downloaded at levels 0 to *i*-1 are put into a thread and perform level *i*-1 pre-loading animation. The (*i*-1)th thread is killed only when it is at the normal termination and the *i*th thread has been started. This is done by using multi-thread concept in Java. The animator starts a bare-bones agent but gradually grows. In any time, there is one animation thread running. The main applet keeps loading frames and extra features. It is only necessary to keep track of loading and frame animation. The interface of the animator is as following:

```
public class animator extends Applet implements Runnable {
    // current animation thread

    private Thread AnimatorThread;

    // two objects for double-buffering

    protected Image offImage;
protected Graphics offGraphics;

// use MediaTracker to synchronise animation

protected MediaTracker tracker;

public void thread_start()

{

// Start a new animation thread when AnimatorThread is null

}

public void thread_stop()

{

// stop animation thread and clean double buffers

}

public void run()

{
\par    // animating images

}

public void paint_style (Graphics g)

{

// four paint styles

}
```

```
// event handle and animation
}
```

Two objects are defined to eliminate flicking using double buffering. MediaTracker is used to track the loading process at each level, and synchronise loading and animation.

Non-Linear Control

There are three ways to achieve non-linear play:

1. *Interactive:*

A user can stop replay, freeze the current frame or step into the previous frame and the next frame using the mouse at the client side. Interactive non-linear speed control can be achieved by dragging the mouse in different speeds. This provides a better way to view the whole animation sequence. The non-linear control is implemented in the following object:

```
public void handle_mouse_event(Event e) {
  switch (e.id) {
    case Event.MOUSE_DOWN:
    // press the mouse to switch between animating and browsing modes
    case Event.MOUSE_UP:
    // resumes animation by tracking the frame the user is viewing
    case Event.MOUSE_DRAG:
    // track mouse, adjust mouse movement and update the frame
}
```

The client can control the animation speed in both reverse and forward orders by simply dragging the mouse backward or forward with different speeds. The speed of animation increases or decreases with the speed of mouse movement, while the frame number is displaying to provide a better way to view the animation. The display of the next frame during the nonlinear control can be expressed as: $f_n = f_c + f_{d_2 - d_1}$, where f_n denotes the next frame number, f_c is the current frame number and $(d_2 - d_1)$ can be calculated from mouse movement.

2. *Duration-tag:*

The animation class is activated from an applet call in a html tag:

```
<APPLET code="animator.class"
    codebase="http://www.cse.unsw.edu.au/~vip"
    width=200 height=230>


<PARAM name="basename"
value="http://www.cse.unsw.edu.au/~vip/baby/T">
<PARAM name="num_images" value="78">
<PARAM name="frame_per_second" value="10">
<PARAM name="extension" value="gif">
</APPLET>
```

Each frame can be attached with a duration tag, eg, <PARAM name="duration_tag" value="2,2,1,3,...,2">, where the numbers define the duration of each frame in milliseconds. There is no extra coding needed for the frames.

3. *Parametrical control:*

The animation can also be controlled by a parametrical function, eg. <PARAM name="velocity" value="sin">, where a sinusoid wave is used to define the velocity of the animation sequence.

Industrial Applicability

The direct application is for animating web objects for the Internet. Although all three non-linear control modes can be used in the web environment, the most useful one is likely to be interactive control. At present, a large part of the popularity of animation is due to its novelty. However, the range of usage available is still quite limited and there is a lack of tools to assist reply and multi-casting.

With this progressive animation in place, it is possible to envisage a boom in the number of animations used in web pages. If the number of animations in the World-Wide-Web is significantly large, installing this animation on popular browsers may be justified.

Besides the application in the Internet, the animation tool can be used for animation control in commercial art studios. Animation has been widely used in TV advertisements, movie production, special visual effects and visual illustrations. The current approach in commercial studios needs an expensive equipment with manual control of play, stop and speed, etc. In computer-assisted design, non-linear editing has been widely accepted, such as Adobe Premiere, Director, etc. Such software will put a high demand on non-linear animation tools. The interactive control in this scheme could replace the current manual control method while the duration-tag method and the parametrical control method will add extra features for non-linear animation control.

It will be appreciated by persons skilled in the art that numerous variations and/or modifications may be made to the invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects as illustrative and not restrictive.

CLAIMS:

1. A method for animating web objects, including the following steps:

(a) dividing an animation sequence comprising an ordered series of frames, into a plurality of strings of different frames, the first string comprising one frame of the sequence, or more frames widely spaced from each other in the sequence, the second string contains one or more frames, and the frames of the second string are widely spaced from each other and from the frame, or between the frames, of the first string, each successive string then contains the same number or more frames than the preceding strings, and the frames of each successive string are widely spaced from each other and between the frames of the immediately preceding string;

(b) transmitting the first string of frames;

(c) receiving the first string of frames and then, using multi-thread controls, displaying a first partial animation sequence comprising the frame, or frames, of the first string in a first thread which will repeat until it is killed;

(d) transmitting and receiving the second string of frames while the first partial animation sequence is being displayed;

(e) then, once the second string is received and the first thread reaches its last frame, killing the first thread and displaying a second partial animation sequence comprising the frames of the first and second string, in sequential order, in a second thread which will repeat until it is killed;

(f) transmitting and receiving the next string of frames while the last partial animation sequence is being displayed;

(g) then, once the next string is received and the last thread reaches its last frame, killing the last thread and displaying a next animation sequence comprising all the frames of the received strings, in sequential order, in a next thread which will repeat until it reaches its last frame and is killed;

(h) repeating steps (f) and (g) until all the strings have been received and the entire animation sequence is being displayed.

2. A method for animating web objects according to claim 1, where the next frame to be displayed is related to the last by some formula dependant on the movement of a pointer such that the animation speed in both reverse and forward orders is controlled by the speed of movement of the pointer.

3. A method for animating web objects according to claim 2, where the display of the next frame during the nonlinear control is expressed as: $f_n = f_c$

$+ f_{d_2 - d_1}$, where f_n denotes the next frame number, f_c is the current frame number and $(d_2 - d_1)$ can be calculated from pointer movement.

4. A method for animating web objects according to claim 1, where a duration tag is attached to each frame to define the duration of each frame.

5. A method for animating web objects according to claim 1, where the animation is controlled by a parametrical function, where a waveform is used to define the velocity of the animation sequence.

6. A system for animating web objects, including:

(a) dividing means to divide an animation sequence comprising an ordered series of frames, into a plurality of strings of different frames, where, the first string comprises one frame of the sequence, or more than one frame widely spaced from each other in the sequence, the second string contains the same number or more frames than the first, and the frames of the second string are widely spaced from each other and from the frame, or between the frames, of the first string, each successive string then contains the same number or more frames than the preceding string, and the frames of each successive string are widely spaced from each other and between the frames of the immediately preceding string;

(b) a transmitter to transmit the strings of frames in order, starting with the first;

(c) a receiver to receive the strings of frames in order, starting with the first;

(d) a multi-thread controller to display a first partial animation sequence comprising the frame, or frames, of the first string in a first thread which will repeat until it reaches its last frame and the controller kills it following reception of the second string, the controller then displays a second partial animation sequence comprising the frames of the first and second string, in sequential order, in a second thread which will repeat until it reaches its last frame and is killed following reception of the next string, then, once each next string is received and the last thread reaches its last frame, the controller kills the last thread and displays a next animation sequence comprising all the frames of the received strings, in sequential order, in a next thread and so on until all the strings are received and the entire animation sequence is being displayed.

7. A system for animating web objects according to claim 6, where the next frame to be displayed is related to the last by some formula dependant

on the movement of a pointer such that the animation speed in both reverse and forward orders is controlled by the speed of movement of the pointer.

8. A system for animating web objects according to claim 7, where the display of the next frame during the nonlinear control is expressed as: $f_n = f_c + f_{d_2 - d_1}$, where f_n denotes the next frame number, f_c is the current frame number and $(d_2 - d_1)$ can be calculated from pointer movement.

9. A system for animating web objects according to claim 6, where a duration tag is attached to each frame to define the duration of each frame.

10. A system for animating web objects according to claim 6, where the animation is controlled by a parametrical function, where a waveform is used to define the velocity of the animation sequence.

11. A transmitter for animating web objects, including:

(a) dividing means to divide an animation sequence comprising an ordered series of frames, into a plurality of strings of different frames, where, the first string comprises one frame of the sequence, or more than one frame widely spaced from each other in the sequence, the second string contains the same number or more frames than the first, and the frames of the second string are widely spaced from each other and from the frame, or between the frames, of the first string, each successive string then contains the same number or more frames than the preceding string, and the frames of each successive string are widely spaced from each other and between the frames of the immediately preceding string;

(b) a transmitter to transmit the strings of frames in order, starting with the first.

12. A method of transmission for animating web objects, including the following steps:

(a) dividing an animation sequence comprising an ordered series of frames, into a plurality of strings of different frames, the first string comprising one frame of the sequence, or more frames widely spaced from each other in the sequence, the second string contains one or more frames, and the frames of the second string are widely spaced from each other and from the frame, or between the frames, of the first string, each successive string then contains the same number or more frames than the preceding strings, and the frames of each successive string are widely spaced from each other and between the frames of the immediately preceding string; and

(b) transmitting the strings of frames in order, starting with the first.

13. A receiver for animating web objects, including:

(a) a receiver to receive strings of frames in order, starting with the first, the strings of frames being divided from an animation sequence comprising an ordered series of frames, the first string comprising one frame of the sequence, or more than one frame widely spaced from each other in the sequence, the second string contains the same number or more frames than the first, and the frames of the second string are widely spaced from each other and from the frame, or between the frames, of the first string, each successive string then contains the same number or more frames than the preceding string, and the frames of each successive string are widely spaced from each other and between the frames of the immediately preceding string;

(b) a multi-thread controller to display a first partial animation sequence comprising the frame, or frames, of the first string in a first thread which will repeat until it reaches its last frame and the controller kills it following reception of the second string, the controller then displays a second partial animation sequence comprising the frames of the first and second string, in sequential order, in a second thread which will repeat until it reaches its last frame and is killed following reception of the next string, then, once each next string is received and the last thread reaches its last frame, the controller kills the last thread and displays a next animation sequence comprising all the frames of the received strings, in sequential order, in a next thread and so on until all the strings are received and the entire animation sequence is being displayed.

14. A receiver according to claim 13, employing client-side buffers, progressive loading frames, and multi-thread animation.

15. A receiver according to claim 13, where the next frame to be displayed is related to the last by some formula dependant on the movement of a pointer such that the animation speed in both reverse and forward orders is controlled by the speed of movement of the pointer.

16. A receiver according to claim 15, where the display of the next frame during the nonlinear control is expressed as: $f_n = f_c + f_{d_2 - d_1}$, where f_n denotes the next frame number, f_c is the current frame number and $(d_2 - d_1)$ can be calculated from pointer movement.

17. A receiver according to claim 13, where a duration tag is attached to each frame to define the duration of each frame.

18. A receiver according to claim 13, where the animation is controlled by a parametrical function, where a waveform is used to define the velocity of the animation sequence.

19. A method of reception for animating web objects, including:

(a) receiving strings of frames in order, starting with the first, the strings of frames being divided from an animation sequence comprising an ordered series of frames, the first string comprising one frame of the sequence, or more than one frame widely spaced from each other in the sequence, the second string contains the same number or more frames than the first, and the frames of the second string are widely spaced from each other and from the frame, or between the frames, of the first string, each successive string then contains the same number or more frames than the preceding string, and the frames of each successive string are widely spaced from each other and between the frames of the immediately preceding string;

(b) displaying a first partial animation sequence comprising the frame, or frames, of the first string in a first thread which will repeat until it reaches its last frame and then killing it following reception of the second string, then displaying a second partial animation sequence comprising the frames of the first and second string, in sequential order, in a second thread which will repeat until it reaches its last frame and then killing it following reception of the next string, then, once each next string is received and the last thread reaches its last frame, killing the last thread and displaying a next animation sequence comprising all the frames of the received strings, in sequential order, in a next thread and so on until all the strings are received and then displaying the entire animation sequence.

20. A method according to claim 19, employing client-side buffers, progressive loading frames, and multi-thread animation.

21. A method according to claim 19, where the next frame to be displayed is related to the last by a formula dependant on the movement of a pointer such that the animation speed in both reverse and forward orders is controlled by the speed of movement of the pointer.

22. A method according to claim 21, where the display of the next frame during the nonlinear control is expressed as: $f_n = f_c + f_{d_2 - d_1}$, where f_n denotes the next frame number, f_c is the current frame number and $(d_2 - d_1)$ can be calculated from pointer movement.

23. A method according to claim 19, where a duration tag is attached to each frame to define the duration of each frame.

24. A method according to claim 19, where the animation is controlled by a parametrical function, where a waveform is used to define the velocity of the animation sequence.

25. A signal for animating web objects, including:
a plurality of strings of different frames divided from an animation sequence comprising an ordered series of frames, where, the first string comprises one frame of the sequence, or more than one frame widely spaced from each other in the sequence, the second string contains the same number or more frames than the first, and the frames of the second string are widely spaced from each other and from the frame, or between the frames, of the first string, each successive string then contains the same number or more frames than the preceding string, and the frames of each successive string are widely spaced from each other and between the frames of the immediately preceding string;

26. A signal according to claim 25, where the signal is divided into 'pyramid' structured stages, and the top layer of the 'pyramid' is the first transmitted with as small amount of data as possible and then the rest of the layers are transmitted progressively through each level.

27. A method of non-linear play for animating web objects, comprising the steps of:
receiving a signal according to claim 25, where the next frame to be displayed is related to the last by a formula dependant on the movement of a pointer such that the animation speed in both reverse and forward orders is controlled by the speed of movement of the pointer.

28. A method according to claim 27, where the display of the next frame is expressed as: $f_n = f_c + f_{d_2 - d_1}$, where f_n denotes the next frame number, f_c is the current frame number and $(d_2 - d_1)$ can be calculated from pointer movement.

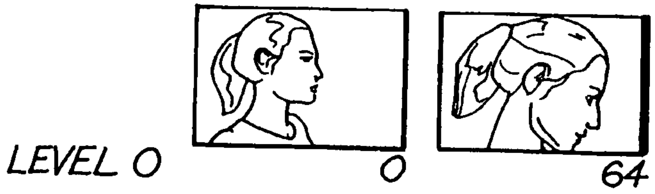


FIG. 1

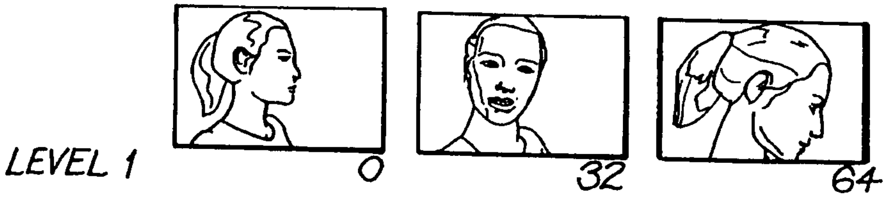


FIG. 2

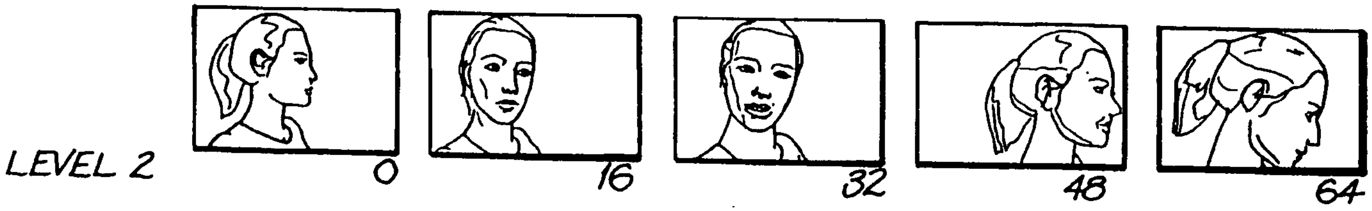


FIG. 3

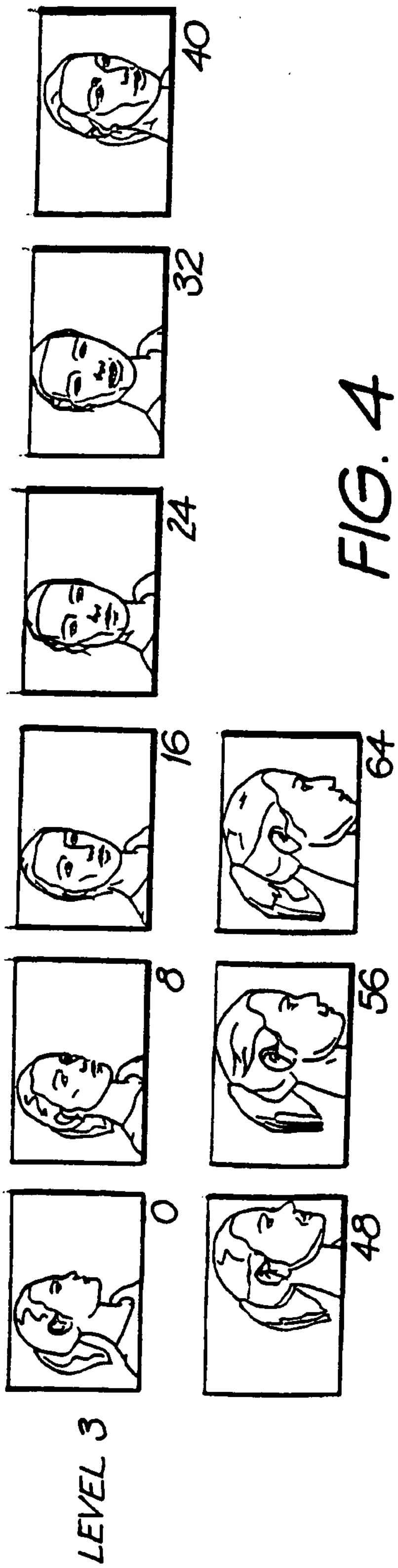

LEVEL 3
0
8
16
24
32
40
48
56
64

FIG. 4

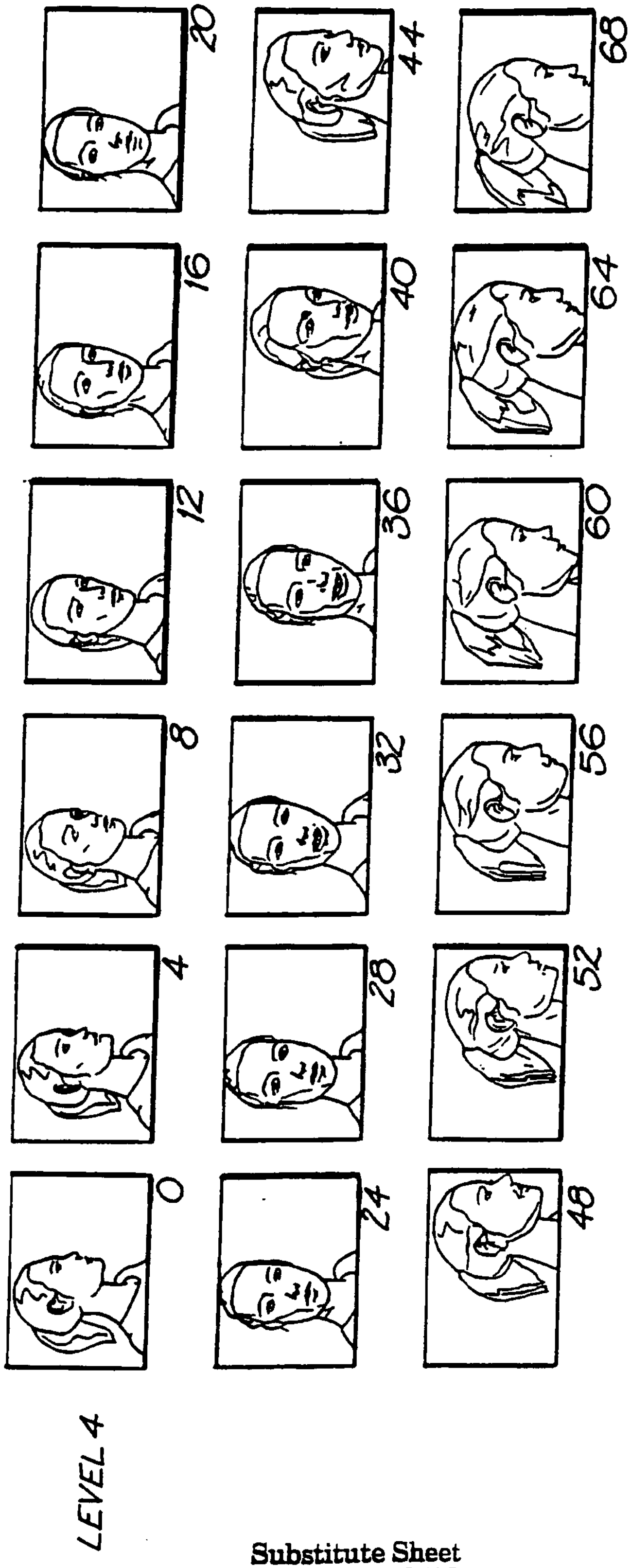

LEVEL 4
0
4
8
12
16
20
24
28
32
36
40
44
48
52
56
60
64
68

FIG. 5

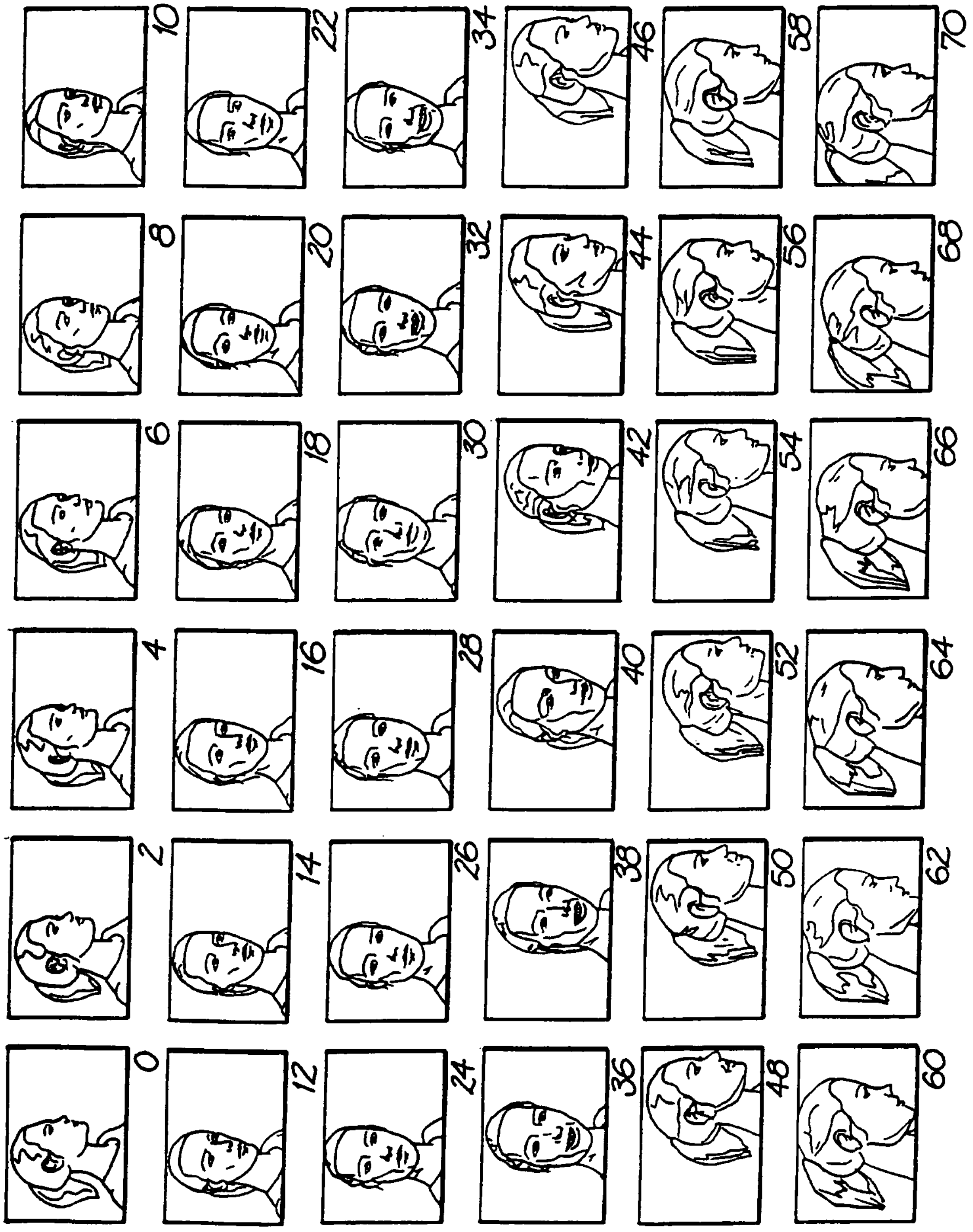

LEVEL 5
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
58
60
62
64
66
68
70

FIG. 6

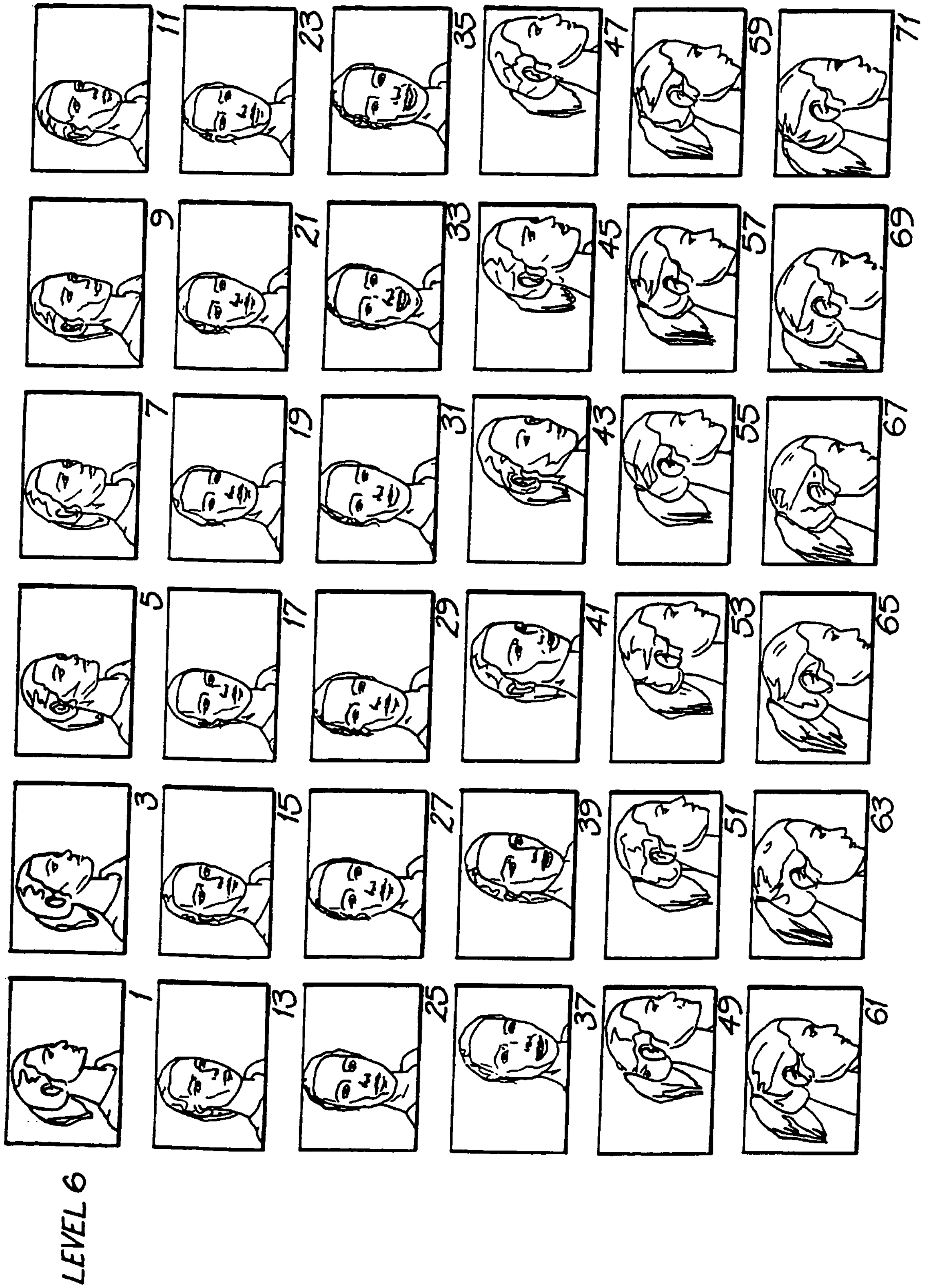

LEVEL 6
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
51
53
55
57
59
61
63
65
67
69
71

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU00/00694

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl. ^{7}: G06T 13/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC: G06F 13/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
WPAT with keywords

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,A	WO 00/11847 (KONINKLIJKE PHILIPS ELECTRONICS N.V.) 2 March 2000	
P,A	WO 99/39264 (MESSAGE MEDIA, INC) 5 August 1999	
P,A	Derwent Abstract Accession No. 2000-030784/03, Class W04, JP 11-298784 A (FUJI PHOTO FILM CO LTD) 29 October 1999	
P,A	Derwent Abstract Accession No. 1999-595798/51, Class T01, JP 11-259669 A (HITACHI LTD) 24 September 1999	

□ Further documents are listed in the continuation of Box C □ See patent family annex

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 10 August 2000	Date of mailing of the international search report 14.08.00
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA E-mail address: pct@ipaustralia.gov.au Facsimile No. (02) 6285 3929	Authorized officer J.W. THOMSON Telephone No : (02) 6283 2214